



Managing Subprograms

- Jayendra Khatod

Objectives

- **Contrast system privileges with object privileges**
- **Contrast invokers rights with definers rights**
- **Identify views in the data dictionary to manage stored objects**
- **Describe how to debug subprograms by using the DBMS_OUTPUT package**

Required Privileges

- A DBA grants System privileges
CREATE (ANY) PROCEDURE
ALTER ANY PROCEDURE
DROP ANY PROCEDURE
EXECUTE ANY PROCEDURE
- The Owner grants Object privileges like **EXECUTE**
- To be able to refer and access objects from a different schema in a subprogram, you must be granted access to the referred objects explicitly, not through a role.

Granting Access to Data

- **Direct access**

```
GRANT SELECT  
ON employees  
TO scott;
```

- **Indirect access**

```
GRANT EXECUTE  
ON query_emp  
TO green;
```

- **The procedure executes with the privileges of the owner (default)**

Using Invoker's-Rights

```
CREATE PROCEDURE query_employee  
(p_id IN emp.empno%TYPE,  
p_name OUT emp.ename%TYPE,  
p_salary OUT emp.sal%TYPE,  
p_comm OUT emp.comm%TYPE)  
AUTHID CURRENT_USER  
IS  
BEGIN  
SELECT ename, sal,  
comm  
INTO p_name, p_salary, p_comm  
FROM emp  
WHERE empno=p_id;  
END query_employee;
```

The procedure now executes with the privileges of the user

Managing Stored PL/SQL Objects

Managing of stored PL/SQL Objects involves the following:

- **Compile errors**
- **General information**
- **Source code**
- **Debug information**
- **Parameters**

Managing Stored PL/SQL Objects

Stored PL/SQL Objects can be managed through the following :

- **USER_OBJECTS Data Dictionary View**
- **USER_SOURCE Data Dictionary View**
- **USER_ERRORS Data Dictionary View**
- **SHOW ERRORS command**
- **DESCRIBE command**
- **The DBMS_OUTPUT supplied package**

USER_OBJECTS

- To obtain the names of all PL/SQL stored objects within a schema, query the **USER_OBJECTS** data dictionary view.
- Also, you should examine the **ALL_OBJECTS** and **DBA_OBJECTS** views, each of which contains the additional column **OWNER**, for the owner of the object.

USER_SOURCE

- To obtain the text of a stored procedure or function, use the USER_SOURCE data dictionary view.
- Also examine the ALL_SOURCE and DBA_SOURCE views, each of which contains the additional column OWNER, for the owner of the object.
- If you have lost the source file, use SQL*Plus to regenerate it from USER_SOURCE, or use the Procedure Builder Stored Program Unit editor.

USER_ERRORS & SHOW_ERRORS

- **Obtaining Compile Errors**
 - To obtain the text for compile errors, use the **USER_ERRORS** data dictionary view or the
 - **SHOW ERRORS SQL*Plus** command.
 - Also examine the **ALL_ERRORS** and **DBA_ERRORS** views, each of which contains the
 - additional column **OWNER**, for the owner of the object.
- **SHOW ERRORS**
 - Use **SHOW ERRORS** without any arguments at the **SQL** prompt to obtain compilation errors for the last object you compiled.

USER_TRIGGERS

- **USER_TRIGGERS**

- **If you have lost the source file, use SQL*Plus to regenerate it from USER_TRIGGERS or use the Procedure Builder Database Trigger Editor.**
- **Also examine the ALL_TRIGGERS and DBA_TRIGGERS views, each of which contains the additional column OWNER, for the owner of the object.**

DESCRIBE Command

- **Describing Procedures and Functions**
 - **To display a procedure or function and its parameter list, use the DESCRIBE SQL*Plus command.**
 - **Example: Display the parameter list for the ADD_DEPT procedure.**
 - **To display the parameter list for the ADD_DEPT procedure, which has defaults. The DEFAULT column only says there is a default value; it does not give the actual value itself.**

DESCRIBE Command

- **SQL> DESCRIBE ADD_DEPT**
- **PROCEDURE ADD_DEPT**
- **Argument Name Type**
In/Out Default
- -----
- **V_NAME VARCHAR2(25) IN DEFAULT**
- **V_REGION_ID NUMBER(7) IN DEFAULT**

Debugging PL/SQL Program

- **The DBMS_OUTPUT supplied package**
 - **Accumulates information into a buffer**
 - **Allows retrieval of the information from the buffer**
 - **Autonomous procedure calls (for example, writing the output to a log table)**
 - **Software that uses DBMS_DEBUG**
 - **Procedure Builder**
 - **Third-party debugging software**

Summary

- **System and object privileges**
- **Definer's model**
- **Direct and Indirect access**
- **Data dictionary views such as USER_SOURCE, USER_OBJECTS, USER_TRIGGERS...etc..**
- **Methods to Debug a PL/SQL Program:**
 - **DBMS_OUTPUT**
 - **GUI Based Procedure Builders**
 - **Autonomous Transaction Procedures**



Thank You !